CLAIMS

1 – Hardware unit for controlling access (20), by a processor (10, 110, 120) to a peripheral (P) of this processor, said hardware unit (20) including:

- means (26) of triggering an interrupt of said processor, termed a control interrupt;

- means (21) of obtaining, from said processor and after said triggering, an access authorisation code (Code-AA) to said peripheral (P);

- means (22) of comparing said access authorisation code (Code-AA) with a predetermined reference value (Code-UMCA); and

- so-called validation means (22, 23, 25) designed to generate an electrical signal (SIG_VAL) to validate an access electrical signal (CS, WE, PWR) to said peripheral (P), depending on the outcome of said comparison.

2 – Access control hardware unit according to claim 1, characterised in that said control interrupt is a non-maskable interrupt (NMI1).

3 – Control hardware unit according to claim 1 or 2, characterised in that it additionally includes means (21) of obtaining a trigger code (Code-DD), and in that said means (26) of triggering said control interrupt (NMI1) are designed to trigger said interrupt following the acquisition of said trigger code (Code-DD).

4 – Access control hardware unit according to claim 3, characterised in that it additionally includes means (22) of comparing said trigger code (Code-DD) with said predetermined reference value (Code-UMCA), and in that said triggering means (26) are designed to trigger said control interrupt (NMI1) depending on the outcome of said comparison.

5 - Access control hardware unit according to any one of claims 1 to 4, characterised in that it includes means (26) of triggering an interrupt of said processor, termed an alarm interrupt, when said access authorisation code (Code-AA) or said trigger code (Code-DD) is different from the predetermined reference value (Code-UMCA).

6 – Access control hardware unit according to claim 5, characterised in that said alarm interrupt is a non-maskable interrupt (NMI2).

7 – Access control hardware unit according to any one of claims 1 to 6, characterised in that said predetermined reference value (Code-UMCA) is a constant.

8 - Access control hardware unit according to any one of claims 1 to 6, characterised in that it includes means (24) of generating said reference value (Code-UMCA) according to a predetermined law.

9 - Access control hardware unit according to claim 8, characterised in that said predetermined reference value (Code-UMCA) is a counter initialised when said hardware unit is switched on (UMCA), and in that, according to said predetermined law, said counter is incremented each time said access authorisation code (Code-AA) is obtained.

10 - Access control hardware unit according to any one of claims 1 to 9, characterised in that said validation means (22, 23, 25) include logic combination means (25) designed to:

- receive an electrical signal requesting access (CS-RQ, WE-RQ) to said peripheral (P);
- receive said validation signal (SIG_VAL); and
- validate said access electrical signal (CS, WE) as a function of a state (RQ_0, RQ_1) of said access request electrical signal (CS-RQ, WE-RQ), a state (VAL_0, VAL_1) of said validation signal, and a logic represented in a truth table (25).

11 - Access control hardware unit according to claim 10, characterised in that it includes means (26) of reading a state (RQ_0, RQ_1) of said access request electrical signal (CS_RQ, WE_RQ), and means (26) of triggering an interrupt of said processor, termed an alarm interrupt (NMI2), preferably non-maskable, as a function of this state (RQ_0, RQ_1) and of said state (VAL_0, VAL_1) of said access validation electrical signal (SIG_VAL).

12 - Access control hardware unit according to any one of claims 1 to 11, characterised in that it includes means (23) of inhibiting said validation signal (SIG_VAL).

13 – Access control hardware unit according to claim 12, characterised in that said inhibiting means (23) are designed to inhibit said validation signal (SIG_VAL) following at least one access to said peripheral (P).

14 – Access control hardware unit according to claim 12 or 13, characterised in that said inhibiting means (23) are designed to inhibit said validation signal (SIG_VAL) after a predetermined delay counted from the generation of said access validation electrical signal (SIG_VAL), or from the acquisition of said access code (Code-AA).

15 - Processor (110) characterised in that it includes:

- an access control hardware unit (20) according to any one of claims 1 to 14;

- means (VECT) of implementing a control interrupt routine (IRT1) designed to obtain said access authorisation code (Code-AA); and

- means (IRT1) of sending said access authorisation code (Code-AA) to said access control hardware unit (20).

16 – Processor according to claim 15, characterised in that said control interrupt routine includes means of reading said access code (Code-AA) from a protected memory.

17 – Processor according to claim 15 or 16, characterised in that it additionally includes means of sending a trigger code (Code-DD) to said access control hardware unit (20).

18 - Processor according to any one of claims 15 to 17, characterised in that said control interrupt routine (IRT1) is designed to generate said access code (Code-AA) according to a predetermined law.

19 - Processor [according to] claim 18, characterised in that said access code (Code-AA) is a counter, [and] in that said predetermined law involves initialising said counter (Code-AA) when said processor (110) is switched on, and incrementing said counter each time said code (Code-AA) is sent to said hardware unit (20).

20 - Processor according to any one of claims 15 to 19, characterised in that it additionally includes means (VECT) of implementing an alarm interrupt routine (IRT2) designed to generate an alert and/or to inhibit the use of said peripheral (P).

21 – Processor according to any one of claims 15 to 20, characterised in that it includes said peripheral (P), the latter in particular being capable of being

selected from a write controller for a boot memory (120) of said processor and a memory management unit (MMU).

22 - Method of controlling access, by a processor (10, 110, 120) to a peripheral (P) of this processor, characterised in that it includes the following steps:

- triggering (E34) an interrupt of said processor, termed control interrupt;
- obtaining (E37), from said processor and after said triggering, an access authorisation code (Code-AA) to said peripheral (P);
- comparing (E38) said access authorisation code (Code-AA) with a predetermined reference value (Code-UMCA);
- generating (E50) an electrical signal (SIG_VAL) validating an access signal (CS, WE, PWR) to said peripheral (P), depending on the outcome of said comparison step (E30).

23 – Access control method according to claim 22, characterised in that said control interrupt is a non-maskable interrupt (NMI1).

24 – Access control method according to claim 22 or 23, characterised in that said triggering step (E34) is performed after a step of obtaining (E25) a trigger code (Code-DD).

25 – Access control method according to claim 24, characterised in that it additionally includes a step (E30) of comparing the trigger code (Code-DD) with said predetermined reference value (Code-UMCA), and in that said triggering step (E34) is performed depending on the outcome of said comparison step (E30).

26 - Access control method according to any one of claims 22 to 25, characterised in that it includes a step (E100) of triggering an interrupt of said processor, termed an alarm interrupt, when said access authorisation code (Code-AA) or said trigger code (Code-DD) is different from the predetermined reference value (Code-UMCA).

27 - Access control method according to claim 26, characterised in that said alarm interrupt is a non-maskable interrupt (NMI2).

28 – Access control method according to any one of claims 22 to 27, characterised in that said predetermined reference value (Code-UMCA) is a constant.

29 – Access control method according to any one of claims 22 to 27, characterised in that it additionally includes a step (E40) of generating said reference value (Code-UMCA) according to a predetermined law.

30 - Access control method according to claim 29, characterised in that said predetermined reference value (Code-UMCA) being a counter, it additionally includes a step (E10) of initialising said counter, said counter being incremented during said generation step (E40).

31 - Access control method according to any one of claims 22 to 30, characterised in that, during said step (E50) of generating the validation signal:

- the state (RQ_0, RQ_1) of an electrical signal (CS-RQ, WE-RQ) requesting access to said peripheral (P) is read;

- the state (VAL_0, VAL_1) of said validation signal (SIG_VAL) is read; and

- said access electrical signal (CS, WE) is validated as a function of said state (RQ_1) of said access request electrical signal (CS_RQ, WE_RQ), of said state (VAL_1) of the validation signal (SIG_VAL), and as a function of a logic rule.

32 - Access control method according to claim 31, characterised in that it includes a step (E20, E36) of reading a state (RQ_0, RQ_1) of said access request electrical signal (CS_RQ, WE_RQ), and a step (E100) of triggering a maskable interrupt of said processor, termed an alarm interrupt, preferably non-maskable (NMI2), as a function of said state (RQ_0, RQ_1) and of said state (VAL_0, VAL_1) of said access validation electrical signal (SIG_VAL).

33 - Access control method according to any one of claims 22 to 32 characterised in that it includes a step (E70) of inhibiting said validation signal (SIG_VAL).

34 – Access control method according to claim 33, characterised in that said inhibiting step (E70) is performed following at least one step (E65) of accessing said peripheral (P).

35 – Access control method according to claim 33 or 34, characterised in that said inhibiting step is performed after a predetermined delay counted from said step (E50) of generating the validation signal (SIG_VAL) or from the step (E25) of obtaining said trigger code (Code-DD).

36 – Method of managing access to a peripheral (P), characterised in that it includes a step of implementing a routine (IRT1) associated with a control interrupt, preferably non-maskable (NMI1), said control routine including a step (E520) of sending an access authorisation code (Code-AA) to an access control hardware unit (20) according to any one of claims 1 to 14.

37 – Method of managing access to a peripheral (P) according to claim 36, characterised in that it includes a step of reading said access code (Code-AA) from a protected memory with a view to said sending.

38 – Method of managing access to a peripheral (P) according to claim 36, characterised in that it includes a step (E510) of generating, according to a predetermined law, an access authorisation code (Code-AA) to said peripheral (P), with a view to said sending.

39 - Method of managing access according to claim 38, characterised in that said access authorisation code (Code-AA) being a counter, it additionally includes a step of initialising said counter (Code-AA), and in that said generation step (E510) consists in incrementing said counter (Code-AA) before each sending (S100) of this code (Code-AA) to said hardware unit (20).

40 - Method of managing access according to any one of claims 36 to 39, characterised in that it additionally includes a step of implementing an alarm interrupt routine (IRT2), said alarm routine including a step of generating an alert and/or inhibiting the use of said peripheral.

41 - Computer program including an instruction (E630) to access a peripheral (P), characterised in that it includes an instruction (E620) to send a trigger code (Code-DD) to an access control hardware unit (20) of said peripheral (P) according to any one of claims 1 to 14, before the execution of said access instruction.

42 - Computer program according to claim 41, characterised in that it additionally includes means of generating said trigger code (Code-DD) according to said predetermined law.

43 – Processor designed to implement an access control method according to any one of claims 22 to 35 and/or a method of managing access according to any one of claims 36 to 40 and/or a computer program according to claim 41 or 42.

44 – Use of an access control hardware unit (20) according to any one of claims 1 to 14, to validate an access signal to a peripheral (P) which can in particular be selected from a screen, a keyboard, a memory, a communications interface controller, a memory management unit (MMU) or a memory protection unit (MPU).